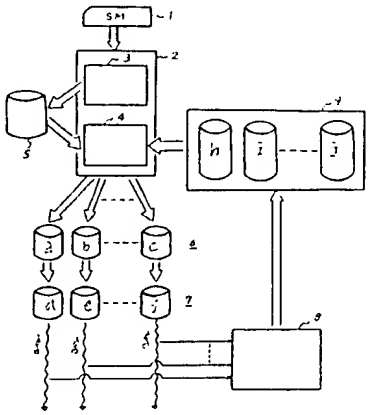
(54) SYSTEM FOR GENERATING [illegible]AMIC INSTRUCTION
(11) 63-276127 (A) (43) 14.11.1988 (19) JP
(21) Appl. No. 62-111572 (22) 7.5.1987
(71) FUJITSU LTD(1) (72) MICHIHIRO NAKAZAWA(1)
(51) Int. Cl⁴. G06F9/44

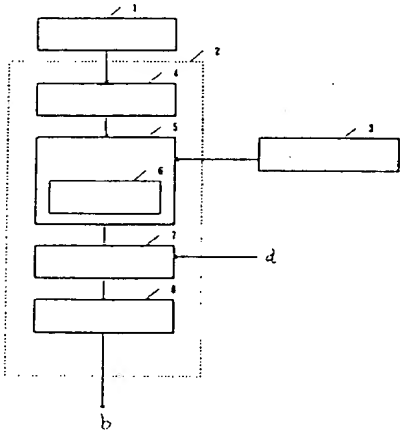


PURPOSE: To improve the executing efficiency of a program by monitoring each instruction executing frequency in an object module at every user and reflecting this monitoring result on the next compiling process for production of an optimum instruction.

CONSTITUTION: When a source module SM 1 is first compiled, a source analyzing part 3 analyzes the SM 1 and an instruction generating part 4 produces an object module (OM) 6 based on only the information collected in a compiling state. An instruction executing frequency counting mechanism 8 works every time each user executes a load module LM 7. These counted frequencies are stored by an instruction executing frequency storing mechanism 9 at every user. When the SM 1 is compiled again by a compiler 2, the SM 1 is analyzed by the part 3 and the part 4 produces the OM 6 while referencing the contents of the compiling information 5 and the executing number of times of the mechanism 9.

a: OM of user 1, b: OM of user 2, c: OM of user n, d: LM of user 1, e: LM of user 2, f: LM of user n, g: execution, h: instruction executing frequency information of user 1, i: instruction executing frequency information of user 2, j: instruction executing frequency information of user n

(54) PROCESSING SYSTEM FOR DECISION OF PROCEDURE NAME BY FILE INPUT
(11) 63-276128 (A) (43) 14.11.1988 (19) JP
(21) Appl. No. 62-110609 (22) 8.5.1987
(71) NEC CORP (72) TAKASHI KANEKO
(51) Int. Cl⁴. G06F9/44

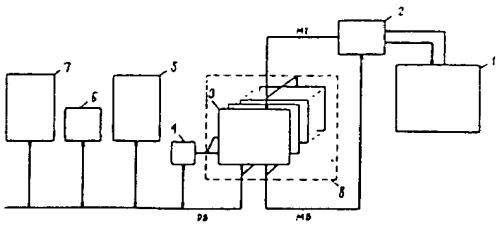


PURPOSE: To decide a setting procedure name as a user procedure name without correcting an original program by inputting a prepared list of procedure names via a file based on the designation of a translation time option.

CONSTITUTION: A translation time option part 1 transmits an option to instruct the reading of a procedure name out of a procedure name list storing part 3 when the original program is translated by a compiler 2. An option analyzing part 4 analyzes the option and a list of procedure names is read out of the part 3 through a designated file to be stored as the user procedure name information 6. An original program analyzing part 7 performs a syntax analysis and the procedure name quotation if included in the information 6 is defined as a user procedure name and as a setting procedure name if not. An object program generating part 8 produces an object program in response to the decided procedure.

5: procedure name file input processing part, a: original program, b: objective program

(54) DATA PROCESSOR
(11) 63-276129 (A) (43) 14.11.1988 (19) JP
(21) Appl. No. 62-111045 (22) 7.5.1987
(71) MATSUSHITA ELECTRIC IND CO LTD (72) RYOICHI WADA(2)
(51) Int. Cl⁴. G06F9/44



PURPOSE: To perform the parallel operations of various lists via plural processing units by transferring the elements of the list data of a table form one by one through a storage device so that they are allocated to those processing units which process said list data.

CONSTITUTION: The data lists of the storage device are transferred one by one by a transfer device 2 so that they are allocated to the processing units 3 respectively. These units 3 can be executed in a single cycle since they perform the shift operations in parallel with each other. The operations of a table form list are carried out to the numeric value and characters by a 2nd arithmetic unit 6 not by the unit 3. The actual value received from the unit 3 is stored in a 2nd storage device 7 and then into a 2nd register device 5 in an operation mode. The operations are carried out between both devices 5 and 6 and these operation results are stored again in the device 7.

1: memory device, 3: condition register, 8: arithmetic unit

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭63－276127

⑤Int.Cl.⁴	識別記号	庁内整理番号	㊸公開 昭和63年(1988)11月14日
G 06 F 9/44	322	A－8724－5B	

審査請求 未請求 発明の数 1 （全6頁）

㊹発明の名称　ダイナミック命令生成方式

㉑特　願 昭62－111572
㉒出　願 昭62(1987)5月7日

㉜発明者　中沢　通太　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内
㉜発明者　松田　政博　福岡県福岡市博多区博多駅前1丁目5番1号　富士通九州通信システム株式会社内
㉛出願人　富士通株式会社　神奈川県川崎市中原区上小田中1015番地
㉛出願人　富士通九州通信システム株式会社　福岡県福岡市博多区博多駅前1丁目5番1号
㉔代理人　弁理士　長谷川　文廣　外1名

明　細　書

1．発明の名称

ダイナミック命令生成方式

2．特許請求の範囲

コンパイラによる命令生成方式において，

原始プログラムからコンパイルされたオブジェクトモジュール中の各命令の実行時の実行回数をカウントする命令実行回数カウント機構（8）と，

この命令実行回数カウント機構（8）の結果を各利用者毎に蓄積していく命令実行回数蓄積機構（9）とを設け，

原始プログラムをコンパイルする時に命令実行回数蓄積機構（9）に蓄積された各利用者の命令実行回数情報を参照して実行回数の多い命令については処理時間を短縮する最適化を行い，命令生成方法を変更することを特徴とするダイナミック命令生成方式。

3．発明の詳細な説明

〔概要〕

本発明は，オブジェクトモジュール中の各命令実行回数を利用者ごとに監視し，その結果を次回のコンパイル処理に反映させて，最適な命令生成を行うことにより，プログラムの実行効率を向上させる。

〔産業上の利用分野〕

本発明は，コンパイラにおける命令生成方式，特に命令生成方法が複数ある場合のコンパイラの最適化処理技術に関する。

〔従来の技術〕

原始プログラムを実行可能なプログラム，即ちOM（Object Module）に翻訳するには，通常コンパイラが用いられる。

第4図は，従来例を示す図である。

第4図において，41はSM（Source Module），42はコンパイラ，43はソース解析部，44は

命令生成部，４５はコンパイル時の情報ファイル，４６はＯＭ(Object Module) である。

ＳＭ (Source Module) ４１は，原始プログラムであり，コンパイラ４２によりコンパイルされて実行可能なＯＭ(Object Module) ４６に翻訳される。

コンパイラ４２は，ＳＭ (Source Module) ４１を実行可能なＯＭ(Object Module) ４６に翻訳する。

ソース解析部４３は，ＳＭ (Source Module) ４１を解析するためのものである。

命令生成部４４は，ＯＭ(Object Module) ４６の命令群を生成するためのものである。

コンパイル時の情報ファイル４５には，コンパイルに必要な情報が蓄積されている。

ＯＭ(Object Module) ４６は，コンパイラ４２により翻訳された実行可能なプログラムである。

原始プログラムＳＭ４１は，コンパイラ４２中のソース解析部４３により命令，文法等が解析され，コンパイル時の情報ファイル４５に蓄積された情報に基づいて，命令生成部４４において命令列が生成され，実行可能なプログラムであるＯＭ４６に翻訳される。

（発明が解決しようとする問題点）

従来のコンパイラによる命令の生成方式では，コンパイル時に収集できる情報のみから命令を生成しているため，実行上最適な命令の生成を行うことができないという問題があった。

例えば，原始プログラムＳＭとして次の例を考える。

```
ＩＦ　Ａ＝Ｂ　ＴＨＥＮ
　　文１
　ＥＬＳＥ
　　文２
ＦＩ
```

この原始プログラムＳＭにおいて，文１と文２が同時に実行されることはない。このため，文１と文２の命令生成順序は任意であるが，文１と文２の実行頻度により出力順序を可変にすれば，実行効率の良いＯＭを生成することができる。

しかしながら，実行頻度は実際に実行しなければ判らないから，従来例のようにコンパイル時に収集できる情報のみから命令を生成するコンパイラでは，実行効率の良いＯＭを生成することはできない。

〔問題点を解決するための手段〕

本発明は，コンパイラによる命令生成方式において，コンパイルされた命令の実行時の命令実行回数をカウントする命令実行回数カウント機構と，この命令実行回数カウント機構の結果を各利用者毎に蓄積していく命令実行回数蓄積機構とを設け，原始プログラムをコンパイルする時に命令実行回数蓄積機構に蓄積された各利用者の命令実行回数情報を反映させた命令生成を行うことにより，命令の実行効率を向上させるものである。

第１図は，本発明の基本構成を示す図である。

第１図において，１はＳＭ (Source Module)，２はコンパイラ，３はソース解析部，４は命令生成部，５はコンパイル時の情報ファイル，６はＯＭ (Object Module)，７はＬＭ (Load Module)，８は命令実行回数カウント機構，９は命令実行回数蓄積機構である。

ＳＭ (Source Module) １は，高級言語で書かれた原始プログラムである。

コンパイラ２は，ソース解析部３及び命令生成部４からなり，ＳＭ１をＯＭ６に翻訳する。

ソース解析部３は，ＳＭ１を解析するためのものである。

命令生成部４は，ＯＭ６の命令群を生成するためのものである。

コンパイル時の情報ファイル５には，コンパイルに必要な情報が蓄積されている。

ＯＭ (Object Module) ６は，コンパイルされた実行可能なプログラムであり，各利用者毎に作成される。

ＬＭ (Load Module) ７は，各利用者がコンピュータにロードして実行中のプログラムである。

命令実行回数カウント機構８は，各利用者がＬ

M7を実行した時のLM7の命令群中の各命令の実行回数をカウントするためのものである。

命令実行回数蓄積機構9は，命令実行回数カウント機構8がカウントした各利用者の命令実行回数を各利用者ごとに蓄積しておくためのものである。

〔作用〕

SM1を最初にコンパイルする時は，ソース解析部3により解析した後，コンパイル時に収集された情報のみに基づいて命令生成部4によりOM6を作成する。

命令実行回数カウント機構8は，各利用者がLM7を実行時に各命令が実行される都度，その回数を数え上げ，その結果は，命令実行回数蓄積機構9が各利用者毎に蓄積していく。

SM1をコンパイラ2により再コンパイルするには，ソース解析部3によりSM1を解析した後，コンパイル時の情報ファイル5に収集された情報に加えて命令実行回数蓄積機構9に蓄積された各利用者の命令実行回数情報をも参照して命令生成部4によりOM6を作成する。

〔実施例〕

第2図は，本発明の1実施例構成を示す図である。

本実施例においては，プログラムの例として，

```
IF  A=B  THEN
      文1
  ELSE
      文2
FI
```

というプログラムを用い，以下これをプログラムAと称する。

第2図において，21はプログラムAのSM(Source Module)，22はコンパイラ，23はソース解析部，24は命令生成部，25はコンパイル時の情報ファイル，26はプログラムAのOM(Object Module)，27はプログラムAのLM(Load Module)，28は命令実行回数カウント機構，29は命令実行回数蓄積機構である。

プログラムAのSM(Source Module)21は，高級言語で書かれたプログラムAの原始プログラムである。

コンパイラ22は，ソース解析部23及び命令生成部24からなり，SM21をOM26に翻訳する。

ソース解析部23は，SM21を解析するためのものである。

命令生成部24は，OM26の命令群を生成するためのものである。

コンパイル時の情報ファイル25には，コンパイルに必要な情報が蓄積されている。

プログラムAのOM(Object Module) 26は，プログラムAのSM21がコンパイルされた実行可能なプログラムである。

プログラムAのLM(Load Module) 27は，利用者がコンピュータにロードして実行中のプログラムである。

命令実行回数カウント機構28は，利用者がLM27を実行した時のLM27の命令群中の各命令の実行回数をカウントするためのものである。

命令実行回数蓄積機構29は，命令実行回数カウント機構28がカウントした命令実行回数を蓄積しておくためのものである。

本実施例において，プログラムAのSM21は，初めは従来例と同じく，コンパイル時の情報25のみに基づいてコンパイルされてOM26となる。OM26は，LM27としてコンピュータにロードされて実行される。この実行時に各命令の実行回数が命令実行回数カウント機構28によりカウントされ，命令実行回数蓄積機構29に蓄積されていく。

次に，プログラムAのSM21を再コンパイルする時には，コンパイル時の情報25と命令実行回数蓄積機構29に蓄積された情報とを参照してコンパイルする。

前掲のプログラムA，即ち，

```
IF  A=B  THEN
      文1
  ELSE
```

文２

Ｆ１

というプログラムを用いて，以下説明する（第３図参照）。

命令実行回数蓄積機構２９に蓄積された情報として文２の実行頻度が高ければ，命令生成部２４の命令生成論理１を用いて，第３図（ａ）に示した命令群を生成する。これにより，図中①で示した分岐命令の実行時間を短縮することができる。

また，命令実行回数蓄積機構２９に蓄積された情報として文１の実行頻度が高ければ，命令生成部２４の命令生成論理２を用いて，第３図（ｂ）に示した命令群を生成する。これにより，図中②で示した分岐命令の実行時間を短縮することができる。

〔発明の効果〕

本発明では，命令生成条件に実行時の各命令の実行頻度を盛り込むので，原始プログラムのコンパイルと実行を繰り返す毎に,より最適な命令生成を実現し，より効率の良いＯＭ(Object Module)を生成することができる。

また，同一の原始プログラムから，各利用者の実行条件に適した，異なったＯＭ(Object Module)を作成し，ソフトウェア製品の実行効率を上げることができる。

４．図面の簡単な説明

第１図は本発明の基本構成を示す図，第２図は本発明の１実施例構成を示す図，第３図は生成されるＯＭの命令群の例を示す図，第４図は従来例を示す図である。

第１図において，

１：ＳＭ (Source Module)

２：コンパイラ

３：ソース解析部

４：命令生成部

５：コンパイル時の情報ファイル

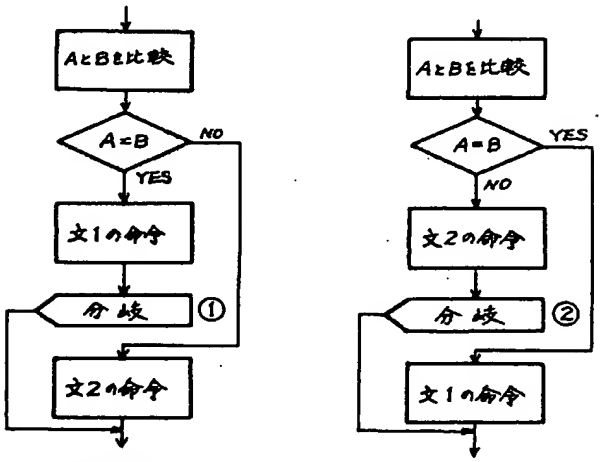
６：ＯＭ (Object Module)

７：ＬＭ (Load Module)

８：命令実行回数カウント機構

９：命令実行回数蓄積機構

特許出願人　富士通株式会社（外１名）
代理人弁理士　長谷川　文廣（外１名）

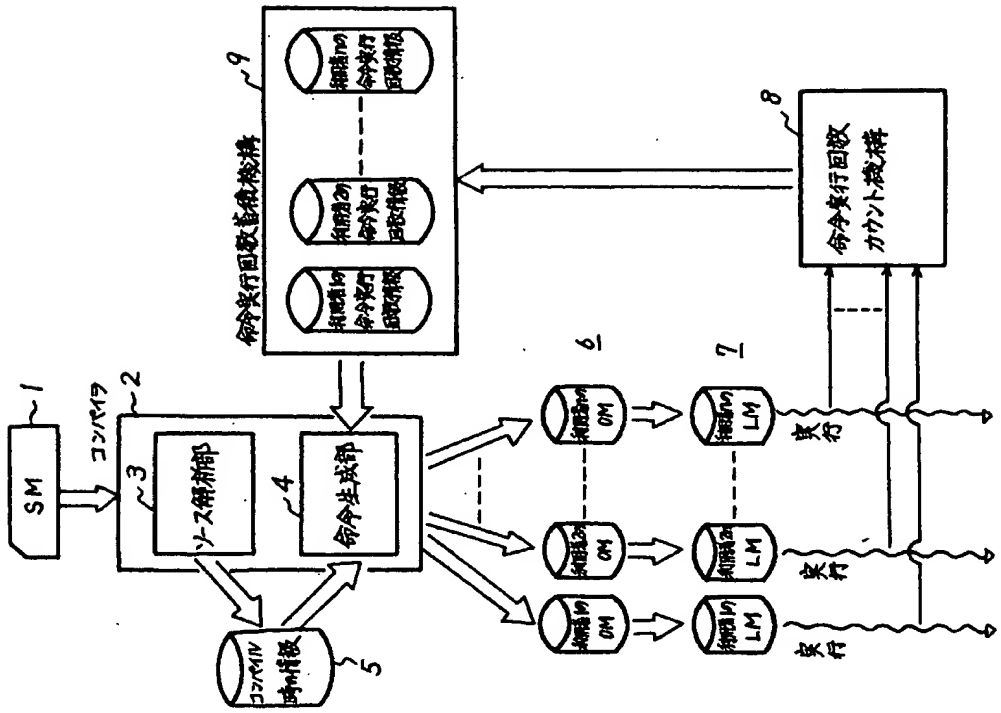


命令生成論理１により生成されるOMの命令群（a）

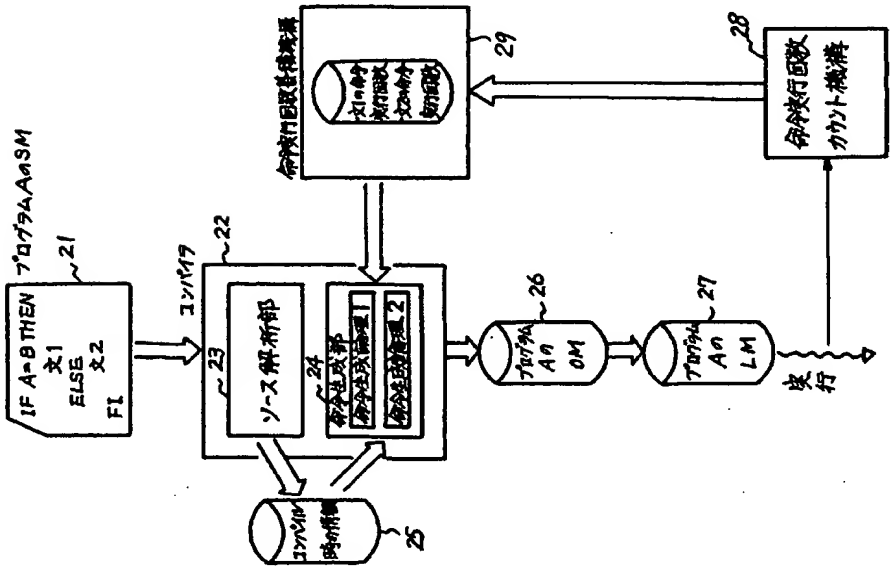
命令生成論理２により生成されるOMの命令群（b）

生成されるOMの命令群の例

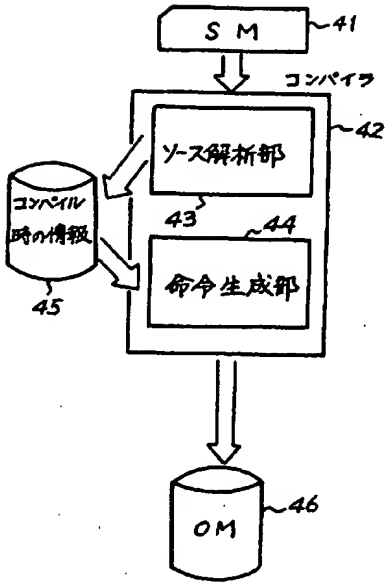
第３図



本発明の基本構成
第 1 図



本発明の1実施例構成
第 2 図



従来例
第4図